Performance, Power, and Resilience Tradeoffs in Algorithmic Design

Modsim 2015

Ian Karlin, Edgar A. Leon, Veselin Dobrev,
Christopher Earl and Tzanio Kolev

August 12, 2015

LLNL-PRES-676255

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Lawrence Livermore National Laboratory

We Do Not Know What the Future Will Bring, but We Have Some Ideas

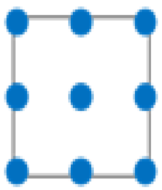
- Future machines will have:
  - Significantly more compute power
  - A lower ratio of memory bandwidth to compute
  - Better performance per watt and higher overall power
  - More complex memory systems
  - Greater parallelism

- Future machines might:
  - Charge for power usage
  - Have hard power constraints
  - Require new resilience techniques

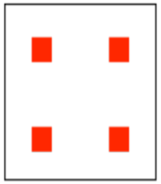
While we do not know what the future will bring flexibility in our applications will be helpful.

BLAST is an Arbitrary Lagrangian Eulerian (ALE) Hydro Code

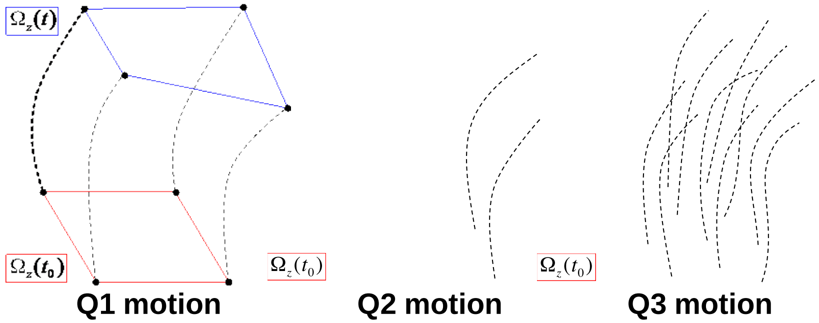
It uses arbitrary order curvilinear finite elements with two centerings

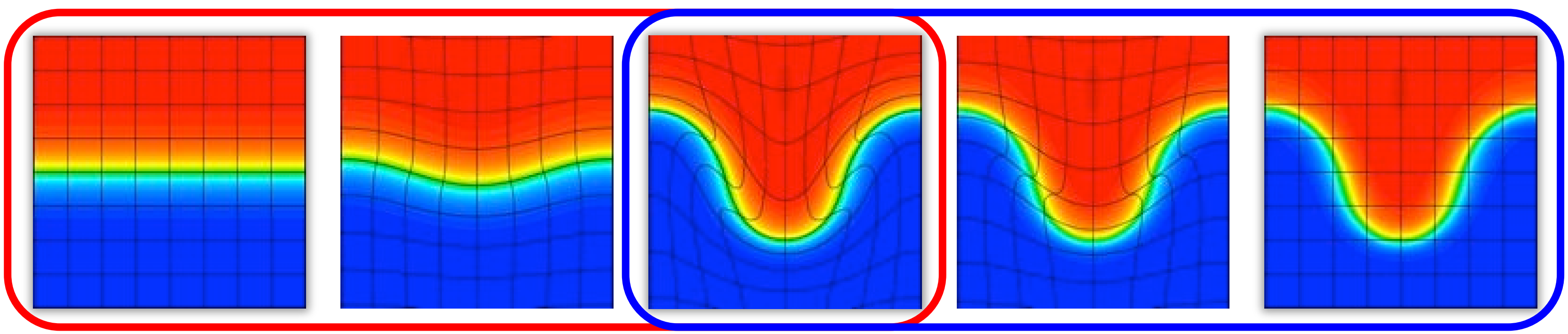


kinematic basis (Q2)



thermodynamic basis (Q1)

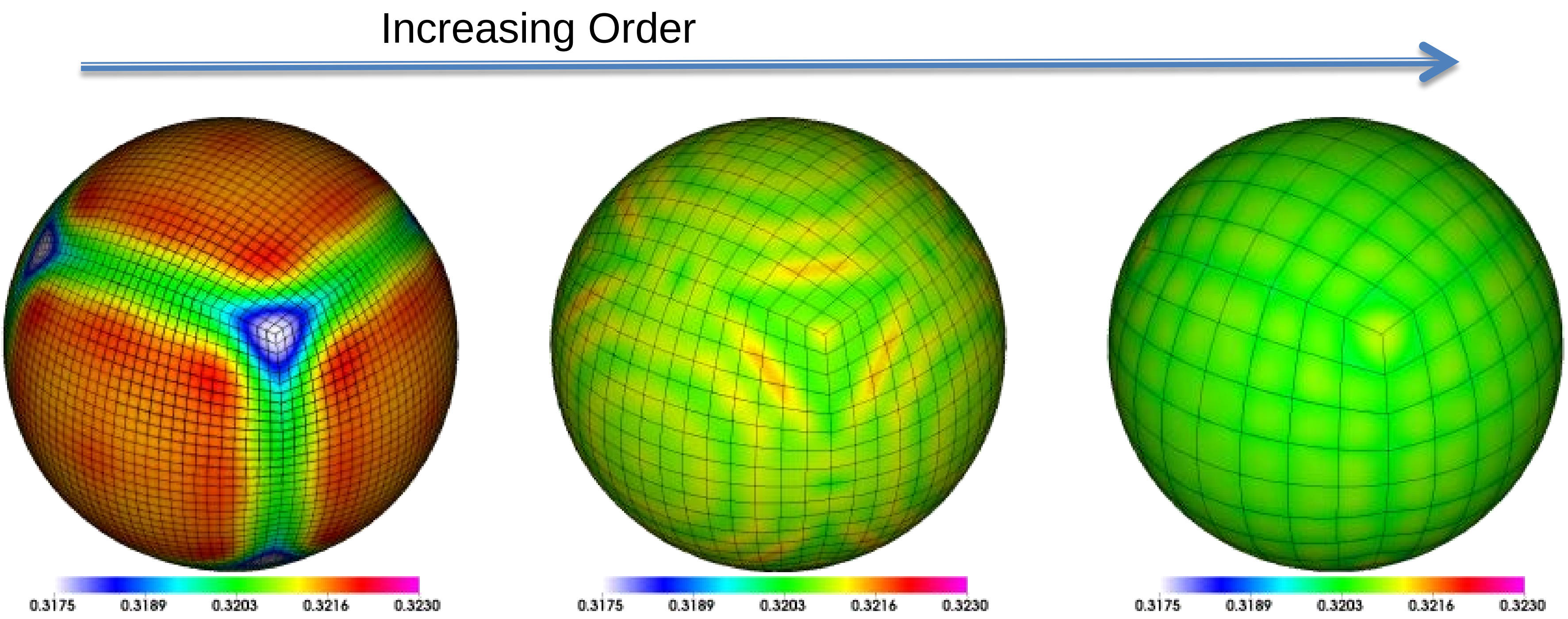




Lagrange Phase

Remap Phase

High-Order Algorithms Produce More Accurate Answers

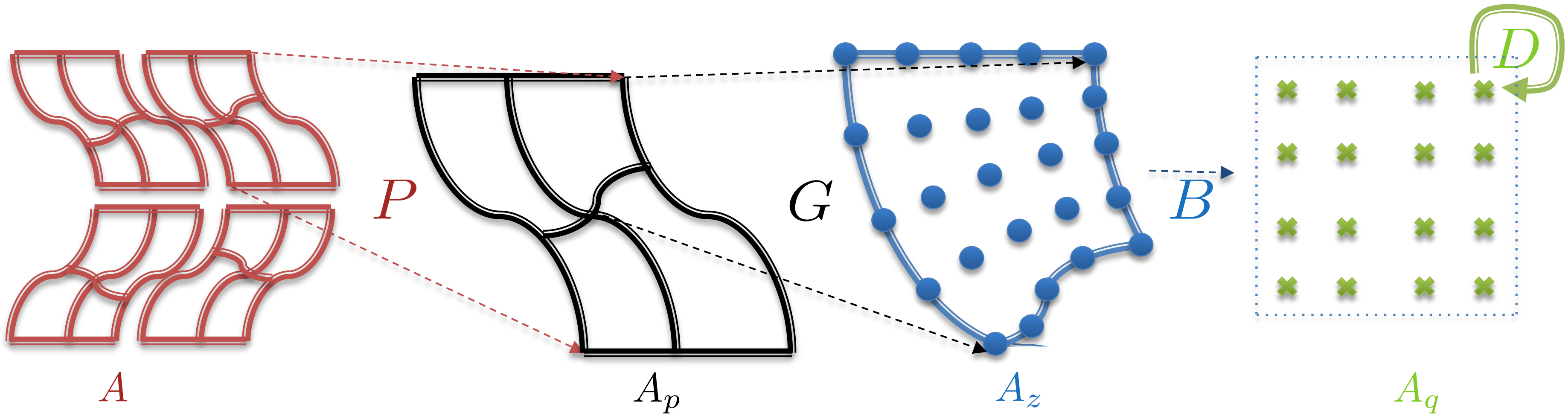


Symmetry is improved as order is increased

Finite Element Codes Produce Multiple Evaluation Options with Different Compute, Bandwidth and Storage Requirements

$$A = P^T G^T B^T D B G P$$

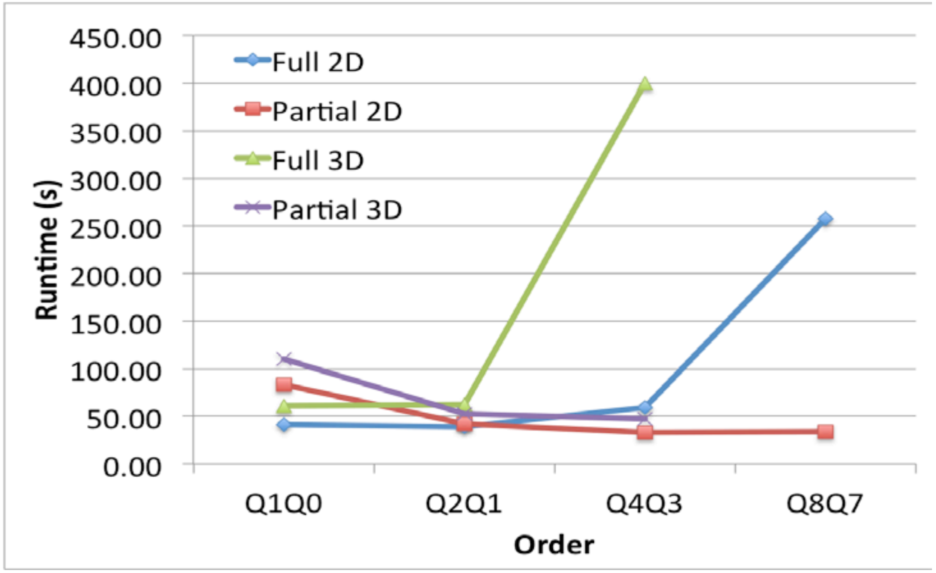
The finite element assembly/evaluation of general bilinear forms (matrices) can be decomposed into *parallel*, *mesh topology*, *FE basis*, and *geometry/physics* components:



Storage options:

1. A_p ,A : standard global matrices (e.g. in CSR/ParCSR format)
2. A_z : local stiffness matrices (e.g. for high-order methods)
3. D : quadrature point data only (independent of basis order)
4. *none* : action-only evaluation (e.g. for explicit methods)

Algorithm Changes Driven by Data Motion Models Led to Performance Improvements



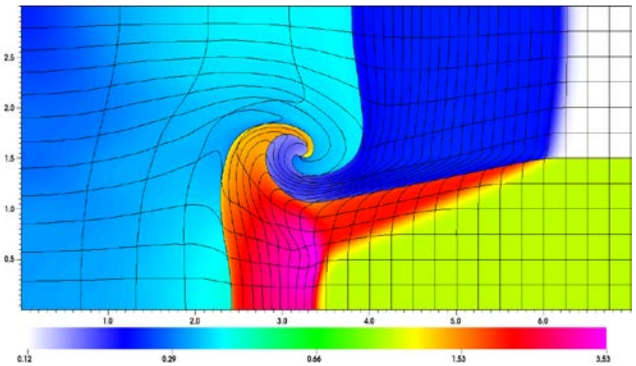
Initial Performance Shows Best Algorithm Varies with Order and Dimension Choice

Models and Tuning Showed Partial Assembly was Better Algorithm When Implemented Right

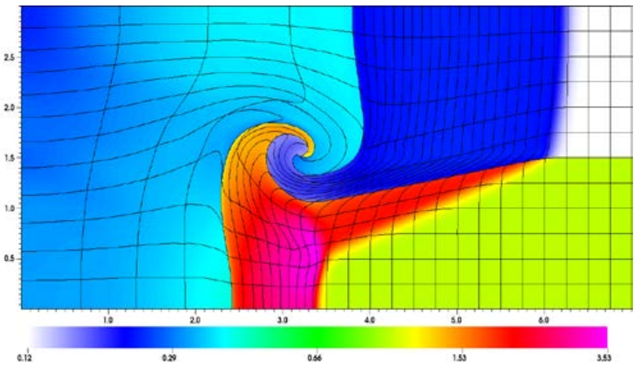
Method	Data Motion Ratio to Ares
Ares 3D	1
Blast Q2Q1	25.28
Blast Q2Q1 Partial	5.34
Blast Q2Q1 Solve	23.28
Blast Q2Q1 Solve Partial	4.10

However, even with our large improvement we still move more data than our competition.

CoDesign Between Mathematicians and Computer Scientists Resulted in Further Improvements



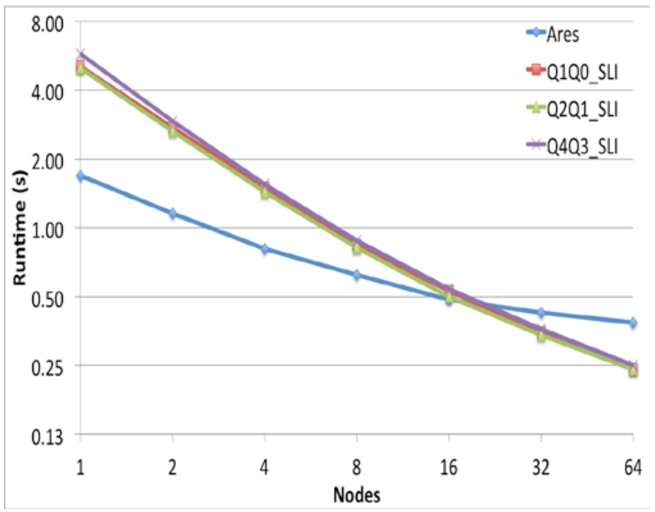
SLI 4 Iterations



PCG Avg 13.7 Iterations

Same Accuracy Less Data Motion

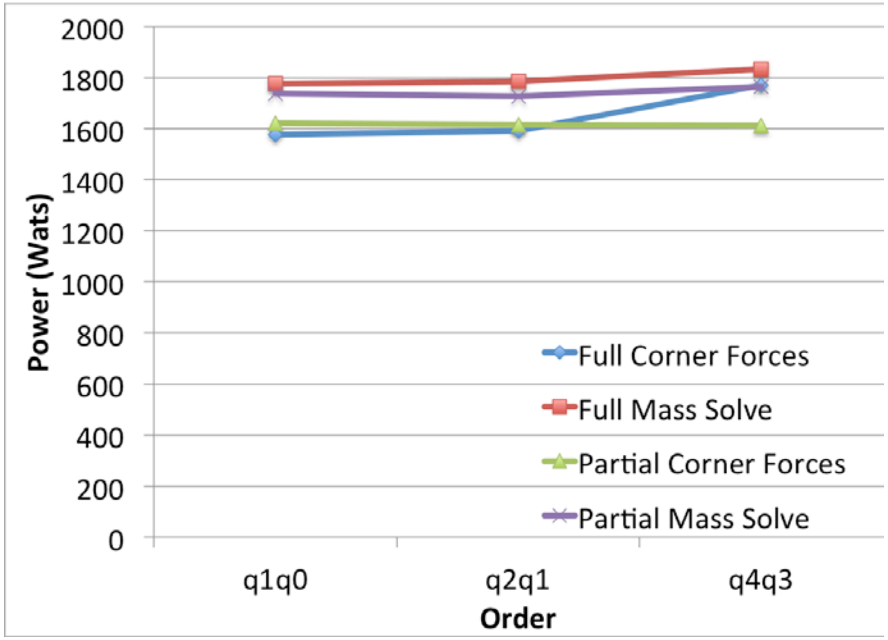
Method	Data Motion Ratio to Ares
Ares 3D	1
Blast Q2Q1	25.28
Blast Q2Q1 SLI	1.74
Blast Q2Q1 Solve	23.28
Blast Q2Q1 SLI Solve	0.45



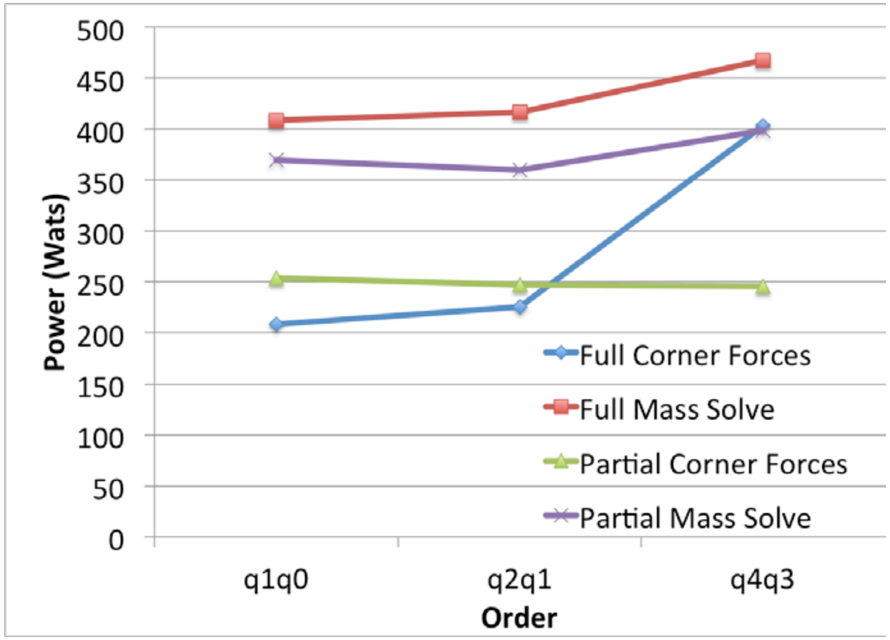
Strong Scaling Results in Better Time to Solution

Partial Assembly Is Also More Power Efficient

Power on 32 Nodes of BG/Q



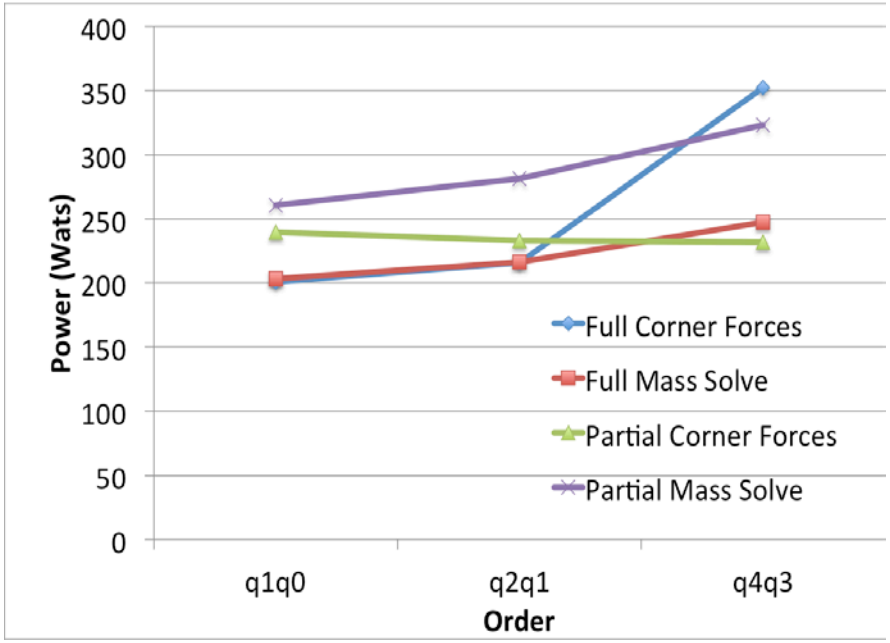
Total Power



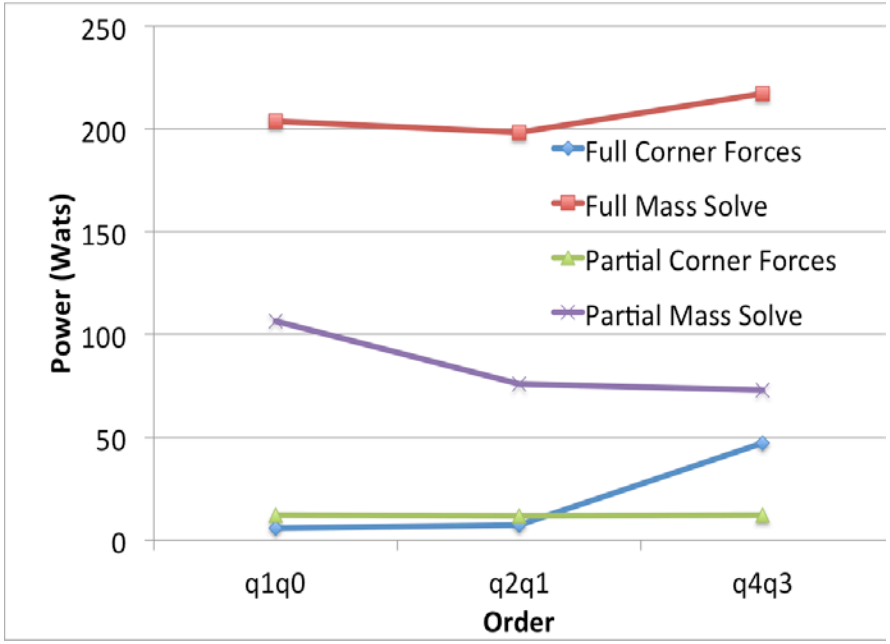
Dynamic Power

Dynamic Power Usage Shifts from DDR to Core

Power on 32 Nodes of BG/Q



Core Dynamic Power

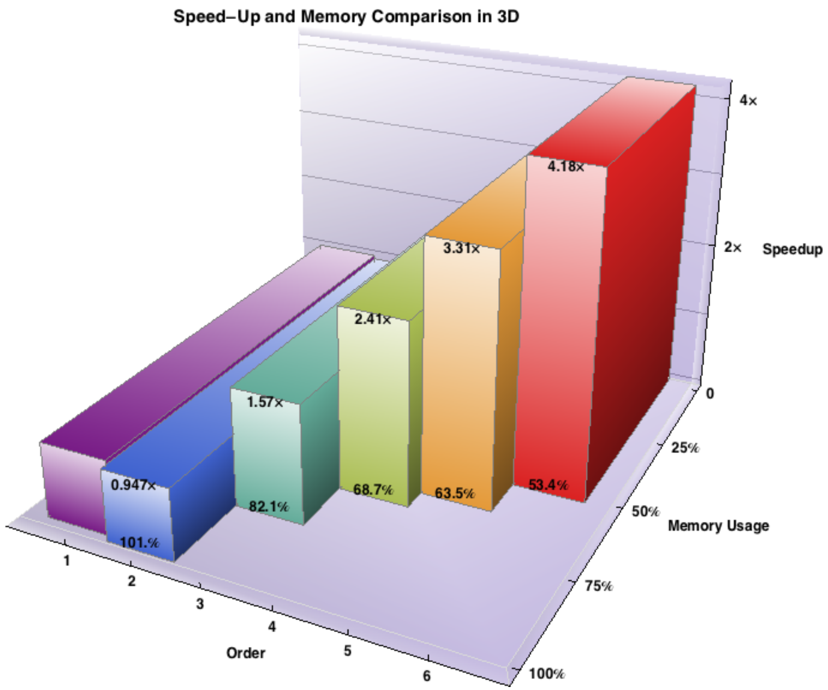


DDR Dynamic Power

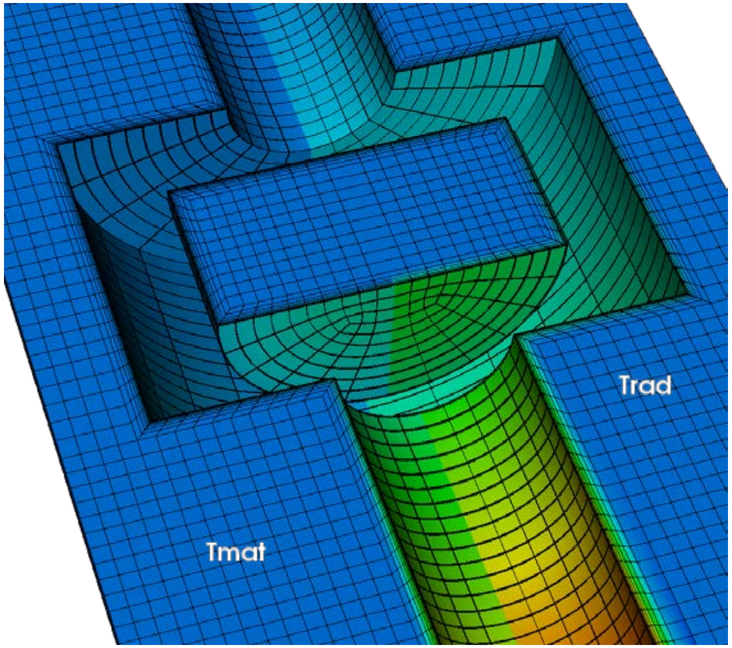
With core power improving faster than DDR power this bodes well for the future.

More Physics and Ways to Solve Them Will Be Added to BLAST Over Time

Static Condensation a Third Option



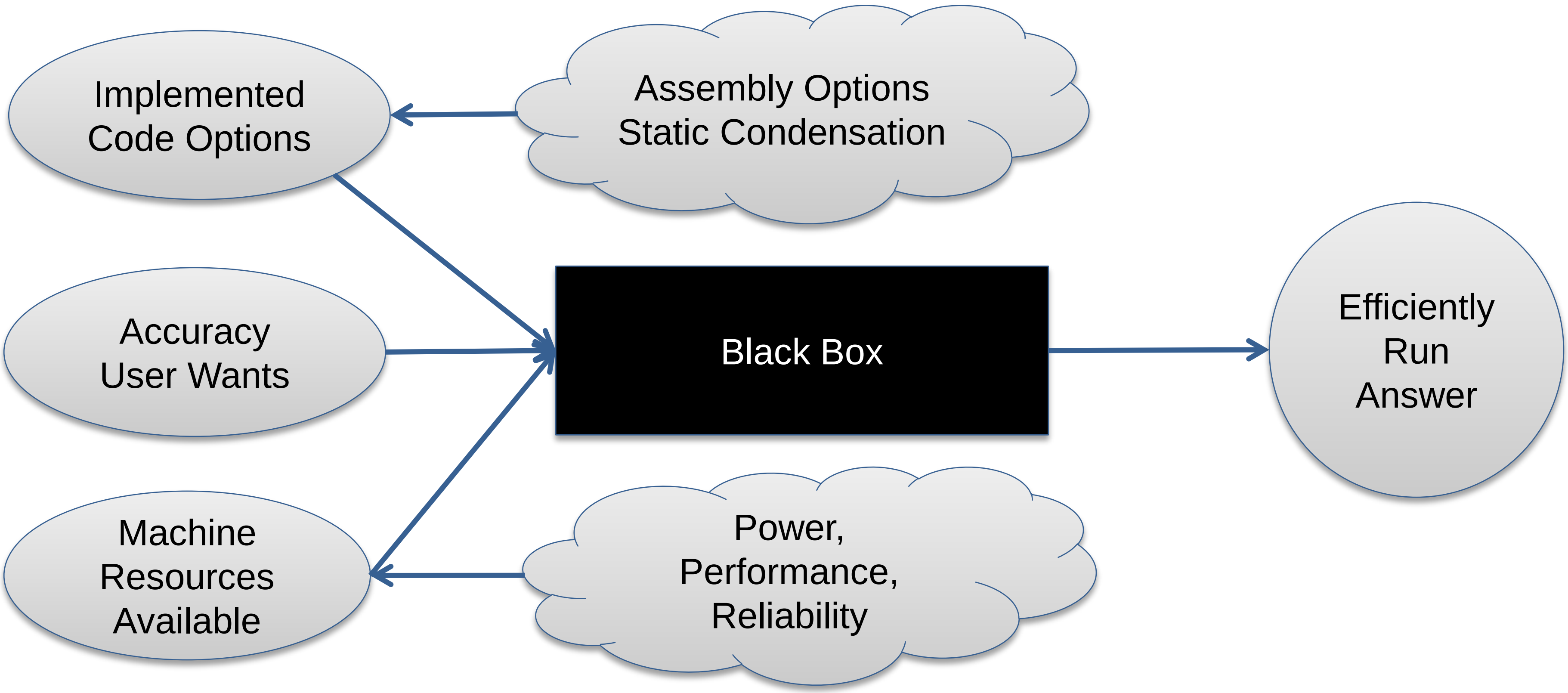
Radiation Diffusion Coupling in Progress



$$H(grad) \xrightarrow{\nabla} H(curl) \xrightarrow{\nabla\times} H(div) \xrightarrow{\nabla\cdot} L_2$$

$H(grad)$	$H(curl)$	$H(div)$	L_2
nodes	edges	faces	zones
Kinematic	(E Field)	Rad Flux	Thermodynamic

How We Would Love to Run Our Code on New Machines in the Future



Steps towards automating this process would boost user and computer center productivity.

Lawrence Livermore National Laboratory